ELSEVIER

Journal of Computational and Applied Mathematics 61 (1995) 165–178

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Simultaneous factorization of a polynomial by rational approximation

Carsten Carstensen[a,*], Tetsuya Sakurai[b]

[a] *Department of Mathematics, Heriot-Watt University, Edinburgh, EH14 4AS, UK*

[b] *Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba 305, Japan*

Received 2 August 1993; revised 8 March 1994

Abstract

In this note we present a numerical method to approximate some relatively prime factors of a polynomial simultaneously. Our approach gives methods of arbitrary order; Grau's method (Carstensen, 1992; Grau, 1971) is obtained as the second order method which is Durand–Kerner's method when we have linear factors. For linear factors our approach yields the simultaneous methods introduced in Sakurai et al. (1991). We prove local convergence and estimate the R-order of the total step version as well as the single step version of the methods. We derive an algorithm and present numerical examples which confirm the convergence behavior theoretically predicted.

Keywords: Polynomial zeros; Factorization of polynomials; Simultaneous methods; Single step methods; Rational approximation

1. Introduction

One of the classical topics in numerical analysis is the computation of polynomial roots. The theory of simultaneous computation of all the zeros of a polynomial, see [16] and the references quoted there, started with Weierstraß in the last century and is highly influenced by using interval arithmetic nowadays. The disadvantage of such methods lies in the fact that they work only for simple roots or for the academic case when we have multiple zeros with a known exact multiplicity. In the presence of clusters, the restrictions on the local convergence results for such methods made the methods useless, i.e., the required accuracy for the starting values to guarantee convergence is so high that we are not interested in improving these approximations in practice even if we have clusters of simple zeros.

* Corresponding author.

SSDI 0377-0427(94)00063-8

Table 1
Related methods

g	Point method	Factoring method
1	[13]	[10]
f'	[15]	[18]
g_j	[17]	(M)

For example, assume that we use a simultaneous method (like Durand–Kerner's) which is developed to compute all the polynomial roots if they are simple. Imagine that we have a polynomial with multiple zeros. Then, the simple approximants cluster near the multiple zero such that their mean is an higher-order approximant of the exact multiple zero [1, 5, 8, 11]. However, the quadratic convergence order of the approximants cannot be observed numerically in this case but we usually see some slow (linear) convergence. One tool for improving the convergence is introduced in [8] and based on the quadratic convergence of the means [5, 8]. Hence, if we have a cluster of zeros, the method in [8] will improve the convergence of the means of the clustering approximants but not the approximations to the zeros.

An alternative tool is the use of simultaneous numerical factorization [4, 6, 9] (see also [5, Section 6]). Given a cluster of zeros $\zeta_j^1, \ldots, \zeta_j^{k_j}$ of a polynomial consider the corresponding factor

$$p_j^*(z) = (z - \zeta_j^1) \cdots (z - \zeta_j^{k_j}) \tag{1}$$

of degree $\deg p_j^* = k_j$. Then compute approximants of the coefficients of p_j^* with respect to some polynomial base (which may be fixed or change in any iteration step, cf. [4]) instead of computing the approximants of the zeros directly. After one has obtained good approximations for the coefficients of p_j^* one can simply compute or estimate the approximations of the zeros of p_j^*.

This note extends Grau's method for simultaneous factoring [9] to arbitrary order by generalizing the method for simultaneous computation of simple polynomial roots [17] and the method for the computation of a single factor [18] which is based on rational approximation of f/g, f being the given polynomial and g being some other polynomial.

To set the method (M) of this paper in an appropriate frame, Table 1 classifies related methods: f is the given polynomial, while f/g is to be approximated. g and some references (of course not a complete list) for the single point method as well as for the factorization method resulting from this approach are shown in Table 1 (using notations from below).

The point methods with $g = 1$ or $g = f'$ compute a single approximation and a second technique for deflation is required to compute all the zeros or factors successively. The other methods in Table 1 compute all the approximants simultaneously. Method [10] belongs to a different class of simultaneous methods than [18] or (M) which are suited here (see [3]).

Note that Newton–Raphson's and Halley's method are particular cases of [13], the Durand–Kerner (or Weierstraß) method and Aberth's method [1] are particular cases of [17], and Grau's method [9] is a particular case of method (M) presented in Section 2.

The rest of this paper is organized as follows: In Section 2 we introduce the method (M) and prove its local feasibility. Asymptotic error estimates are shown in Section 3 which give local

convergence of Q-order $M + 1$. The single step mode of the method is also under consideration there. In Section 4 we derive an algorithm to evaluate the methods numerically. Some numerical examples in Sections 5 and 6 confirm our theoretical results with a combined method taking into account the problem "How to find initial values"? A few remarks in Section 7 conclude this paper.

2. The method

Let f be a monic complex polynomial of degree n having the zeros

$$\zeta_1^1, \dots, \zeta_1^{k_1}, \zeta_2^1, \dots, \zeta_m^1, \dots, \zeta_m^{k_m},$$

with $k_1 + k_2 + \cdots + k_m = n$, $m, k_1, \dots, k_m$ being natural numbers, $m \geqslant 2$, such that

$$\{\zeta_i^1, \dots, \zeta_i^{k_i}\} \cap \{\zeta_j^1, \dots, \zeta_j^{k_j}\} = \emptyset \quad (i, j = 1, \dots, m;\ i \neq j). \tag{2}$$

Note that, with p_j^* from (1),

$$f = p_1^* \cdots p_m^* \tag{3}$$

and, due to (2), $p_1^*, \dots, p_m^*$ are pairwise relatively prime. For convenient notations with upper and lower indices let

$$I := \{(i, j):\ i = 1, \dots, m;\ j = 1, \dots, k_i\}$$

denote the admissible index pairs.

In order to approximate the exact factors $p_1^*, \dots, p_m^*$ (cf. (1)) assume that we are given approximating factors $p_1, \dots, p_m$ with exact degree, i.e., p_j is a monic polynomial of degree k_j $(j = 1, \dots, m)$.

Remark 2.1. Imagine that the zeros of p_j^* (as defined in (1)) define the jth cluster such that

$$\min_{(i,j),(\nu,\mu) \in I,\ i \neq \nu} |\zeta_i^j - \zeta_\nu^\mu|,$$

namely the smallest distance between two different clusters, is much greater than the greatest diameter,

$$\max_{(i,j),(i,k) \in I} |\zeta_i^j - \zeta_i^k|,$$

of one cluster. Note that (2) is satisfied in this case that this represents the practical situation in which we have a perturbed or exact multiple root. As is explained in Section 1, some complex simultaneous methods give a cluster of approximants which define an approximating factor p_j. p_j is the monic polynomial with the clustering approximants as zeros (counting multiplicities).

One step of the presented method of order $M + 1$, M being a natural number, consists in computing a monic polynomial $\hat{p}_j$ of degree $\deg \hat{p}_j = k_j$. $\hat{p}_j$ is determined such that $\hat{p}_j/\hat{q}_j$ is the *Rational Hermite Interpolant* of f/g_j, $g_j := \prod_{i=1,\, i \neq j}^m p_i$, $\hat{q}_j$ being a polynomial of degree $\deg \hat{q}_j \leqslant k_j \cdot (M - 1) - 1$ (for $M \geqslant 2$, $\hat{q}_j \equiv 1$ if $M = 1$) where the interpolation points are the $M \cdot k_j$ zeros of p_j^M, counting multiplicities $(j = 1, \dots, m)$.

In order to be more precise we introduce further notations. Let $\mathbb{P}$ be the vector space of all polynomials and let $\mathbb{P}_k$ be the subspace of all polynomials of degree less than or equal to k while $\mathbb{P}_k^{\text{monic}}$ denotes the monic polynomials with degree k, k being a nonnegative integer. For any polynomial $p(z) = a_0 + a_1 z + \cdots + a_k z^k$ let

$$\|p\| := \max_{i=0,\ldots,k} |a_k|$$

be the norm of p. Hence $(\mathbb{P}, \|\cdot\|)$ is a normed linear space. In this note all polynomials have bounded degrees such that any other norm in $\mathbb{P}$ is essentially equivalent to $\|\cdot\|$. Thus, $\mathcal{U}$ is a neighborhood of a polynomial $p \in \mathbb{P}_k^{\text{monic}}$ in $\mathbb{P}_k^{\text{monic}}$ if there exists $\varepsilon > 0$ with

$$\{q \in \mathbb{P}_k^{\text{monic}}: \|p - q\| < \varepsilon\} \subseteq \mathcal{U}.$$

We start by proving that method (M) is feasible.

Lemma 2.2. *Assuming that $f \in \mathbb{P}_n^{\text{monic}}$ satisfies* (1)–(3) *for any $M \geqslant 1$ and for any $j \in \{1,\ldots,m\}$ there exists some neighborhood $\mathcal{U}_j$ of p_j^* in $\mathbb{P}_{k_j}^{\text{monic}}$ such that* (i)–(iii) *holds.*

(i) *For any $(p_1,\ldots,p_m) \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_m$ the polynomials $p_1,\ldots,p_m$ are pairwise relatively prime and there exist unique polynomials $\hat{p}_j \in \mathbb{P}_{k_j}^{\text{monic}}$ and $\hat{q}_j$ such that $\deg \hat{q}_j \leqslant (M-1)\cdot k_j - 1$ if $M \geqslant 2$ and $\hat{q}_j \equiv 1$ if $M = 1$, and*

$$\hat{q}_j \cdot f \equiv \hat{p}_j \cdot \prod_{k=1, k\neq j}^{m} p_k \pmod{p_j^M} \quad (j = 1,\ldots,m). \tag{4}$$

(ii) *There exists a constant $c_1 > 0$ such that for all $(p_1,\ldots,p_m) \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_m$ we have*

$$\left| [\zeta_i^1,\ldots,\zeta_i^j] \frac{1}{\prod_{k=1,\ k\neq i}^{m} p_k} \right| \leqslant c_1 \quad ((i,j) \in I),$$

where $[\zeta_i^1,\ldots,\zeta_i^j]h$ denotes the divided difference of h with respect to the knots $\zeta_i^1,\ldots,\zeta_i^j$ and $(\zeta_i^j: (i,j) \in I)$ are the zeros of f (cf. (1), (3)).

(iii) *There exists a constant $c_2 > 0$ such that for all $(p_1,\ldots,p_m) \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_m$ we have*

$$\|\hat{r}_j\| \leqslant c_2 \cdot \max_{i=1,\ldots,m; i\neq j} \|p_i^* - p_i\| \quad (j = 1,\ldots,m),$$

where $\hat{r}_j$ is a polynomial uniquely determined in (i) *by*

$$\hat{q}_j \cdot f - \hat{p}_j \cdot \prod_{k=1,k\neq j}^{m} p_k = \hat{r}_j \cdot p_j^M.$$

Proof. Since the zeros of a polynomial depend continuously on its coefficients and we have (2) we conclude that for sufficiently small neighborhoods of $p_1^*,\ldots,p_m^*$ the polynomials $p_1,\ldots,p_m$ are pairwise relatively prime. This proves (ii) as well.

Note that (4) is equivalent to the linearized rational interpolation condition where the data function f/g_j, writing $g_j := \prod_{i=1, i\neq j}^{m} p_i$, is interpolated by a rational function with the polynomials $\hat{p}_j$ and $\hat{q}_j$ with $\deg \hat{p}_j \leqslant k_j$ and $\deg \hat{q}_j \leqslant (M-1)k_j - 1$. The Mk_j interpolation knots are determined

as the zeros of p_j^M, counting multiplicities. Note that we have already seen that f/g_j has no pole at the zeros of p_j^M. The normalization used is that the leading coefficient of $\hat{p}_j$ is equal to one. It is well known that a rational interpolant of this type always exists (Mk_j interpolation conditions for $k_j + (M-1)k_j$ coefficients) but the normalization and the uniqueness have to be treated with a brief look at the linearized rational Hermite interpolation problem, i.e., the Newton–Padé approximation problem.

For the moment let $m' := k_j$ and $n' := k_j(M-1) - 1$ and let $\xi_0, \dots, \xi_{m'+n'}$ be the Mk_j zeros of p_j^M, counting multiplicities, ordered such that $\xi_{l \cdot m'}, \dots, \xi_{(l+1)m'-1}$ are the zeros of p_j ($l = 0, \dots, M$). Let $c_{\mu,\nu}$ denote the divided difference of f/g_j with respect to the knots $\xi_\mu, \xi_{\mu+1}, \dots, \xi_\nu$ if $\mu \leqslant \nu$ and $c_{\mu,\nu} := 0$ if $\mu > \nu$, and define $\omega_k \in \mathbb{P}_k^{\text{monic}}$ by $\omega_k(z) := (z-\xi_0)\cdots(z-\xi_{k-1})$.

We refer, e.g., to [7, Eq. (3.16)] which shows that the interpolation conditions are equivalent to

$$\sum_{\mu=0}^{n'} c_{\mu,\nu} b_\mu = a_\nu \quad (\nu = 0, \dots, m'), \tag{5}$$

$$\sum_{\mu=0}^{n'} c_{\mu,\nu} b_\mu = 0 \quad (\nu = m'+1, \dots, m'+n') \tag{6}$$

and $\hat{p}_j = a_0 + a_1\omega_1 + \cdots + a_{m'}\omega_{m'}$ and $\hat{q}_j = b_0 + b_1\omega_1 + \cdots + b_{n'}\omega_{n'}$.

We have a closer look at (5), (6) with $p_1, \dots, p_m$ replaced by $p_1^*, \dots, p_m^*$. Note that, in this case, $f/g_j = p_j^*$ gives $c_{\mu,\nu} \neq 0$ iff $\nu = \mu + m'$. Then, (5), (6) are equivalent to $b_{n'} = \cdots = b_1 = 0$ and $a_0 = \cdots = a_{m'-1} = 0$ and $a_{m'} = b_0$. Thus, we have unique polynomials $\hat{p}_j \in \mathcal{U}_j$ and $\hat{q}_j$ as provided in (i). For convenient reference below let ω_j^* denote the polynomial ω_j in the present case where $\xi_0, \dots, \xi_{m'+n'}$ are the zeros of $(p_j^*)^M$.

Next we return to the general case and assume that $\varepsilon_1, \dots, \varepsilon_m$ are small,

$$\varepsilon_i := \| p_i - p_i^* \| \quad (i = 1, \dots, m).$$

Note that f/g_j as well as $(c_{\mu,\nu})$ depend continuously on $p_1, \dots, p_m$ such that (5), (6) are equivalent to

$$\begin{pmatrix} o(1) \\ I + o(1) \end{pmatrix} (b_0, \dots, b_{n'})^T = (a_0, \dots, a_{m'}, 0, \dots, 0)^T,$$

where I is the $(n'+1) \times (n'+1)$ unit matrix and the Landau symbol $o(1)$ is a matrix of sufficient dimension (which is not always the same although it is frequently used) with coefficients which tend towards zero if $\varepsilon := \max\{\varepsilon_1, \dots, \varepsilon_m\}$ tends towards zero. Hence, choosing ε sufficiently small we get that (5), (6) leads to a solution (unique by a normalization $b_0 = 1$, say) with $b_i = o(1)$ for $i = 1, \dots, n'$ and $a_i = o(1)$ for $i = 0, \dots, m'-1$ while $a_{m'} = 1 + o(1)$. Observe that ω_i depends also on $p_1, \dots, p_m$ and that $\| \omega_i - \omega_i^* \| = o(1)$. Dividing by $a_{m'}$ we get unique $\hat{p}_j \in \mathcal{U}_j$ and $\hat{q}_j$ satisfying the conditions in (i).

By definition of $\hat{r}_j$ we have $\deg \hat{r}_j \leqslant n - k_j - 1$ (also for $M = 1$). Consequently, a polynomial interpolation where the knots are the zeros of g_j, counting multiplicities, is exact. This leads to polynomials $G_{j,i} \in \mathbb{P}_{k_i - 1}$ uniquely defined by

$$\hat{r}_j = \sum_{i=1, i\neq j}^{m} G_{j,i} \cdot \prod_{\mu=1, i\neq\mu\neq j}^{m} p_\mu.$$

Let $x_i^1, \dots, x_i^{k_i}$ be the zeros of p_i. Then, for $(i,k) \in I$ with $i \neq j$ we have

$$[x_i^1, \dots, x_i^k]\left(G_{j,i} \cdot \prod_{\mu=1, i\neq\mu\neq j}^{m} p_\mu \right) = [x_i^1, \dots, x_i^k](p_i^* g_i^* \cdot \hat{q}_j / p_j^M).$$

Because of (ii) and because $\|\hat{q}_j\|$ is bounded for ε being sufficiently small (as seen above) this proves by induction on $k = 1, 2, \dots, k_i$ that

$$[x_i^1, \dots, x_i^k] G_{j,i} = O([x_i^1, \dots, x_i^k] p_i^*)$$

is bounded uniformly for any $p_1, \dots, p_m$ in a small neighborhood of $p_1^*, \dots, p_m^*$. We used the Landau symbol $O(\cdot)$ defined by $h_1 = O(h_2)$ iff $|h_1|/|h_2|$ is bounded uniformly for all $(p_1, \dots, p_m) \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_m$. Since $\|p_i^* - p_i\| = \varepsilon_i$ we also have that $[x_i^1, \dots, x_i^k] G_{j,i} = O(\varepsilon_i)$ $(k = 1, \dots, k_i)$, whence

$$\|G_{j,i}\| = O(\varepsilon_i).$$

Using this in the above interpolation representation of $\hat{r}_j$ proves

$$\hat{r}_j = O\left(\max_{i=1,\dots,m; i\neq j} \varepsilon_i \right). \qquad \square$$

We are now in the position to state our method (M).

Total step method (M). Assume that $f \in \mathbb{P}_n^{\text{monic}}$ satisfies (1)–(3) and that $\mathcal{U}_1, \dots, \mathcal{U}_m$ satisfy (i)–(iii) of Lemma 2.2. Then, for $M \geqslant 1$ and for any $(p_1, \dots, p_m) \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_m$ one step of method (M) reads

$$(p_1, \dots, p_m) \mapsto (\hat{p}_1, \dots, \hat{p}_m),$$

where $(\hat{p}_1, \dots, \hat{p}_m)$ has to be computed in total step mode (TS) with $\hat{p}_j$ satisfying the condition (i) in Lemma 2.2.

Remark 2.3. If $M = 1$ the method (M) is due to Grau [9] and the results of this note generalize the results of [4] (for $M = 1$ only) to arbitrary order $M + 1$.

3. Local convergence

The asymptotic convergence is estimated in the following theorem. The case $M = 1$ was treated in [4].

Theorem 3.1. *Under the assumptions of Lemma 2.2 there exists a constant $c_3 > 0$ (depending only on $\mathcal{U}_1, \dots, \mathcal{U}_m$) such that for one step of method* (M) *holds*

$$\|p_j^* - \hat{p}_j\| \leqslant c_3 \cdot \|p_j^* - p_j\|^M \cdot \max_{k=1,\dots,m; k\neq j} \|p_k^* - p_k\| \quad (j = 1, \dots, m). \tag{7}$$

Proof. Fix $(p_1, \dots, p_m) \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_m$ and fix $j \in \{1, \dots, m\}$. From (4) we have a polynomial $\hat{r}_j$ with

$$\frac{\hat{q}_j \cdot f - p_j^M \hat{r}_j}{g_j} = \hat{p}_j,$$

where $\prod_{k=1,k\neq j}^{m} p_k := g_j$. Letting $g_j^* := \prod_{k=1,k\neq j}^{m} p_k^* = f/p_j^*$, this gives

$$\hat{p}_j - p_j^* = p_j^* \left(\frac{\hat{q}_j g_j^*}{g_j} - 1 \right) - \frac{p_j^M r_j}{g_j}.$$

Let $[\zeta_j^1, \ldots, \zeta_j^k]h$ denote the divided difference of h with respect to the knots $\zeta_j^1, \ldots, \zeta_j^k$, $k \in \{1, \ldots, k_j\}$, $(\zeta_i^j : (i,j) \in I)$ being the zeros of f, cf. (1), (3). Then,

$$[\zeta_j^1, \ldots, \zeta_j^k](p_j^* - \hat{p}_j) = [\zeta_j^1, \ldots, \zeta_j^k]\left(p_j^M \frac{\hat{r}_j}{g_j} \right).$$

Let $\varepsilon_j := \| p_j^* - p_j \|$. Using Leibniz's rule we have to deal with three kinds of divided differences $[\zeta_j^i, \ldots, \zeta_j^k]$. First,

$$[\zeta_j^i, \ldots, \zeta_j^k] p_j^M = [\zeta_j^i, \ldots, \zeta_j^k](p_j - p_j^*)^M = O(\varepsilon_j^M).$$

Secondly, because of Lemma 2.2(iii),

$$[\zeta_j^i, \ldots, \zeta_j^k]\hat{r}_j = O\left(\max_{\mu=1,\ldots,m;\mu\neq j} \varepsilon_\mu \right).$$

Thirdly, because of Lemma 2.2(ii),

$$[\zeta_j^i, \ldots, \zeta_j^k](1/g_j) = O(1).$$

Thus,

$$[\zeta_j^1, \ldots, \zeta_j^k](p_j^* - \hat{p}_j) = O\left(\varepsilon_j^M \cdot \max_{\mu=1,\ldots,m;\mu\neq j} \varepsilon_\mu \right),$$

which concludes the proof. □

Remark 3.2. By Theorem 3.1 we have that, given sufficiently good approximating factors $p_1, \ldots, p_m$, the method (M) is feasible, i.e., any generated vector of approximating factors lies in $\mathcal{U}_1 \times \cdots \times \mathcal{U}_m$ as well, and the iterative process is convergent with Q-order $M + 1$. Moreover, we have Q-order M in each factor which means that if the initial approximating factors are sufficiently good then the iteration in which only k of the m factors are improved is locally convergent of Q-order M.

So far we considered the *total step mode* of the method (M). Next we consider its *single step mode*. *Single step method* (M). Assume that $f \in \mathbb{P}_n^{\text{monic}}$ satisfies (1)–(3) and that $\mathcal{U}_1, \ldots, \mathcal{U}_m$ satisfy (i)–(iii) of Lemma 2.2. Then, for $M \geqslant 1$ and for any $(p_1, \ldots, p_m) \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_m$ one step of method (M) reads

$$(p_1, \ldots, p_m) \mapsto (\hat{p}_1, \ldots, \hat{p}_m),$$

where $(\hat{p}_1, \ldots, \hat{p}_m)$ has to be computed in single step mode (SS) with $\hat{p}_j$ satisfying the condition (i) in Lemma 2.2 with (4) replaced by

$$\hat{q}_j \cdot f \equiv \hat{p}_j \cdot \prod_{k=1}^{j-1} \hat{p}_k \cdot \prod_{k=j+1}^{m} p_k \pmod{p_j^M} \quad (j = 1, \ldots, m). \tag{8}$$

Thus $\hat{p}_1, \ldots, \hat{p}_{j-1}$ has to be computed before $\hat{p}_j$.

Table 2
$M + \rho_{m,M}$ for some m and M

$m \backslash M$	1	2	3	4	5
2	2.618	4.000	5.302	6.561	7.791
3	2.324	3.521	4.671	5.796	6.904
4	2.220	3.353	4.452	5.533	6.603
5	2.167	3.267	4.341	5.401	6.451
6	2.134	3.214	4.273	5.321	6.361
7	2.112	3.179	4.228	5.267	6.300
8	2.096	3.154	4.196	5.229	6.257
9	2.085	3.135	4.171	5.201	6.225
10	2.075	3.120	4.153	5.178	6.200

We refer to e.g. [14] for the definition of the R-order of convergence. Since we only know that the R-order of the total step mode is (at least) $M + 1$, the next theorem predicts a faster convergence of the single step mode as in [2].

Theorem 3.3. *Assume that* $f \in \mathbb{R}_n^{\text{monic}}$ *satisfies* (1)–(3) *and that we have sufficiently good initial approximating factors* $p_1, \ldots, p_m$. *Then, the single step mode of method* (M) *is feasible, i.e., any generated vector of approximating factors lies in* $\mathcal{U}_1 \times \cdots \times \mathcal{U}_m$ *as well, and the iterative process is convergent with* R-*order of convergence*

$$O_R(M) \geqslant M + \rho_{m,M},$$

where $\rho_{m,M} > 1$ *is the unique positive root of* $\rho^m - \rho - M = 0$.

Proof. By Theorem 3.1, the assertion follows from [16, Theorem 2.4]. □

Some values for the bound of the R-order of convergence $M + \rho_{m,M}$ of Theorem 3.3 are given in Table 2 (truncated to four digits) for varying values of M (increasing with the columns) and m (increasing with the rows).

4. Practical realization aspects

For the calculation of the new approximating factors $\hat{p}_1, \ldots, \hat{p}_m$ from the given approximating factors $p_1, \ldots, p_m$, we need some algorithm which calculates the rational Hermite interpolant $\hat{p}_j/\hat{q}_j$ of f/g_j. We refer to [18] for such an algorithm. When $M = 1$, the algorithm to calculate $\hat{p}_j$ is slightly modified because $\hat{p}_j/\hat{q}_j$ satisfying (4) with $M = 1$ is not the rational Hermite interpolant. let $\hat{h}_j$ be the interpolant for f/g_j of which the interpolation points are k_j zeros of p_j. Then $\hat{p}_j = p_j + \hat{h}_j$ satisfies (4).

Note that g_j is given by the product of p_i, i.e., $g_j := \prod_{l \neq j} p_i$, and expansion of this product causes large computational costs. Thus we use the Hermite interpolant $\hat{g}_j$ such that

$$g_j := \prod_{i=1, i \neq j}^{m} p_i \equiv \hat{g}_j \,(\mathrm{mod}\, p_j^M)$$

instead of g_j.

We now summarize the algorithm.

Step 1: Calculate the Hermite interpolant $\hat{g}_j$ of $g_j = \prod_{i=1, i \neq j}^{m} p_i$ such that

$$g_j \equiv \hat{g}_j \,(\mathrm{mod}\, p_j^M) \quad \text{and} \quad \deg \hat{g}_j \leqslant M \cdot k_j - 1.$$

Step 2: Calculate the Hermite interpolant $\hat{h}_j$ of $h_j := f/\hat{g}_j$ such that

$$f \equiv \hat{g}_j \cdot \hat{h}_j \,(\mathrm{mod}\, p_j^M) \quad \text{and} \quad \deg \hat{h}_j \leqslant M \cdot k_j - 1.$$

Step 3: In case of $M = 1$, let $\hat{p}_j := p_j + \hat{h}_j$. In case of $M \geqslant 2$, calculate the polynomial remainder sequence of p_j^M and $\hat{h}_j$ until the degree of the remainder polynomial is equal to k_j. By using the extended Euclidean algorithm, we can get the polynomials $\hat{p}_j$, $\hat{q}_j$ and $\hat{s}_j$ which satisfy

$$\hat{s}_j \cdot p_j^M + \hat{q}_j \cdot \hat{h}_j = \hat{p}_j,$$

$$\deg \hat{p}_j = k_j, \quad \deg \hat{q}_j \leqslant \deg p_j^M - \deg \hat{p}_j - 1.$$

By computing these steps for $j = 1, \ldots, m$, we get the next approximating factors $\hat{p}_1, \ldots, \hat{p}_m$. For the single step mode, g_j is replaced by

$$g_j := \prod_{i=1}^{j-1} \hat{p}_i \cdot \prod_{i=j+1}^{m} p_i.$$

5. First numerical examples

The examples of this section confirm our theoretical results for the total step and single step modes. The calculations were performed in Mathematica with long-precision arithmetic. Note that real polynomials require only real arithmetic.

The example is taken from [9, Example 1] for comparison. The coefficients of the polynomial are given in [9, Table 1], the polynomial is given by

$$f(z) = p_1^* \cdot p_2^* \cdot p_3^* \cdot p_4^* \cdot p_5^*,$$

where

$$p_1^* = z^2 + 19z + 90, \qquad p_2^* = z^2 + 15z + 56, \qquad p_3^* = z^2 + 11z + 30,$$

$$p_4^* = z^2 + 7z + 12, \qquad p_5^* = z^2 + 3z + 2.$$

Table 3
Error in total step model

M	v	e_1	e_2	e_3	e_4	e_5
1	0	−2.00	−2.00	−2.00	−2.00	−2.00
	1	−1.61	−2.45	−2.58	−3.00	−4.19
	2	−3.98	−4.42	−4.90	−5.86	−7.99
	3	−8.82	−9.09	−10.00	−11.86	−15.34
2	0	−2.00	−2.00	−2.00	−2.00	−2.00
	1	−2.98	−3.31	−3.76	−4.47	−6.08
	2	−10.34	−10.49	−11.70	−13.79	−17.93
	3	−33.02	−33.06	−36.20	−42.37	−50.05
3	0	−2.00	−2.00	−2.00	−2.00	−2.00
	1	−4.42	−4.56	−5.19	−6.16	−8.32
	2	−19.69	−19.79	−22.05	−25.71	−32.87
	3	−82.07	−82.14	−89.97	−102.26	−123.62

Table 4
Error in single step mode

M	v	e_1	e_2	e_3	e_4	e_5
1	0	−2.00	−2.00	−2.00	−2.00	−2.00
	1	−1.61	−1.95	−2.47	−3.37	−5.01
	2	−5.39	−6.70	−7.25	−8.25	−12.19
	3	−12.35	−14.38	−16.68	−20.38	−26.37
2	0	−2.00	−2.00	−2.00	−2.00	−2.00
	1	−2.98	−3.52	−4.29	−5.89	−9.70
	2	−11.04	−12.46	−14.62	−21.06	−33.97
	3	−38.15	−42.26	−51.83	−76.51	−109.69
3	0	−2.00	−2.00	−2.00	−2.00	−2.00
	1	−4.42	−5.12	−6.22	−7.67	−14.20
	2	−20.36	−22.77	−26.41	−37.37	−68.25
	3	−87.85	−97.85	−118.32	−181.24	−297.95

The initial factors were obtained by adding small perturbations to the coefficients of the factors,

$$p_j = p_j^* + 0.01 + 0.01z \quad (j = 1, \ldots, 5).$$

The errors

$$e_j := \log_{10} \| p_j - p_j^* \| \quad (j = 1, \ldots, 5)$$

in step v are shown in Tables 3 and 4 for the total and single step modes, respectively.

6. A combined method

To emphasise practical relevance we describe a combined method dealing with a polynomial which results from a polynomial with multiple zeros by perturbing its coefficients (cf. Section 1 and [5, Section 6]). The polynomial is given by

$$f(z) = p_1^* \cdot p_2^* \cdot p_3^* \cdot p_4^*,$$

where $\delta = 10^{-k}$, $k = 1, 3, 5, 7, 9$, is a small real parameter

$$p_1^* = (z+1)^2 + \delta(1+z), \qquad p_2^* = (z+i)^3 + \delta(1+z+z^2),$$

$$p_3^* = (z+5i)^2 + \delta(1+z), \qquad p_4^* = (z-5i)^2 + \delta(1+z).$$

Note that the zeros are clustering such that theoretically we have simple zeros but any root-finding algorithm which requires simple zeros or zeros with (known) exact multiplicity is expected to fail.

Using a good starting value method (M) behaves well as in the previous example. But how to get good initial values in practice? To give some ideas to this we performed three stages combining the single point methods (i.e., $m = n$, $k_j = 1$, in the above notation; there are more efficient but mathematically equivalent formulae in the literature, see, e.g., [16] and the references quoted there) with the factorization method (M) as follows.

Initial values for the point method. Use Aberth's initial values (cf. [1, 11]): $z_j = g_0 + r \cdot \exp(2\pi \mathrm{i}(j-1)/n + 1/(2n))$ where $g_0 := -0.22 - 0.33\mathrm{i}$, $r = 10.53$, $n = 9$, and set $p_j(z) := z - z_j$.

Perform the single point method (M). Compute iteratively new linear factors $p_j(z) = z - z_j$ by method (M) which is mathematically equivalent to Durand–Kerner's or Weierstraß' method ($M = 1$) or to Aberth's method ($M = 2$) until the following termination criterion is satisfied.

Stopping criterion for the first stage. Terminate the previous computation once the actual approximating factors satisfy

$$\| f \ (\mathrm{mod}\, p_j) \| < \varepsilon_1 \cdot \| f \| \quad \text{for all } j = 1, \ldots, n,$$

where $\varepsilon_1 > 0$ is small — but not too small (we set $\varepsilon_1 = 1/100$).

Form clusters for the second stage. Let $p_j(z) = z - z_j$ be the factors satisfying the stopping criterion for the first stage. Then partition the related distinct approximating zeros $z_1, \ldots, z_n$ in m sets $z_j^1, \ldots, z_j^{k_j}$ $(j = 1, \ldots, m)$ such that

$$\max_{j=1}^{m} \max_{\mu,\nu=1}^{k_j} |z_j^\nu - z_j^\mu| \ll \min_{j,l=1, j\neq l}^{m} \min_{\mu=1}^{k_j} \min_{\nu=1}^{k_l} |z_j^\mu - z_l^\nu|.$$

(For simplicity, we implemented a procedure which gives $\max_{j=1}^{m} \max_{\mu,\nu=1}^{k_j} |z_j^\nu - z_j^\mu| < \frac{1}{2}$.) Set

$$p_j(z) := (z - z_j^1) \cdots (z - z_j^{k_j}) \quad \text{for all } j = 1, \ldots, m.$$

Perform the factoring method (M). Compute iteratively new factors p_j by method (M) until the following termination criterion is satisfied.

Table 5
Single point method: CPU time in msecs (number of iterations)

$M\backslash\delta$	10^{-3}	10^{-5}	10^{-7}	10^{-9}
1	131 (27)	151 (32)	164 (32)	175 (32)
2	112 (18)	129 (21)	133 (19)	144 (19)
3	119 (13)	138 (15)	149 (15)	161 (15)

Table 6
Combined method: CPU time in msecs (number of iterations)

$M\backslash\delta$	10^{-3}	10^{-5}	10^{-7}	10^{-9}
1	116 (20)	118 (20)	118 (20)	118 (20)
2	101 (11)	101 (11)	101 (11)	101 (11)
3	119 (10)	120 (10)	120 (10)	120 (10)

Stopping criterion for the second stage. Terminate the previous computation once the actual approximating factors satisfy

$$\| f \ (\mathrm{mod}\, p_j) \| < \varepsilon_2 \| F \ (\mathrm{mod}\, P_j) \| \quad \text{for all } j = 1, \ldots, m,$$

where $\varepsilon_2 > 0$ is small (we set $\varepsilon_2 = 10^{-12}$) and the polynomials F and P_j are given by f and p_j taking the corresponding moduli of the coefficients instead of the coefficients themselves: if $f(z) = z^n + a_{n-1} \cdot z^{n-1} + \cdots + a_0$ and $p_j(z) = z^{k_j} + b_{k_j-1} \cdot z^{k_j-1} + \cdots + b_0$ then $F(z) := z^n + |a_{n-1}| \cdot z^{n-1} + \cdots + |a_0|$ and $P_j(z) := z^{k_j} - |b_{k_j-1}| \cdot z^{k_j-1} - \cdots - |b_0|$.

The combined method and the single point method were performed in FORTRAN on Macintosh with IEEE double precision with about 16 decimal digits. Table 5 shows the CPU time in milliseconds and in parentheses the related number of iterations used until termination of the single point method with $M = 1, 2, 3$ for a varying polynomial f, i.e., for various parameters δ.

One observes from Table 5 that the smaller the perturbation of the zeros in the cluster are the larger is the computer time of the single point method which affirms considerations in the literature, e.g., in [1, 5, 8].

Table 6 shows the corresponding values for the combined methods (1), (2) and (3) and proves that δ has no practical influence on the computer effort required — as predicted by Theorem 3.1 because there is no difference in the underlying concept of a cluster or a multiple root. The computer time used for stage 2 of the combined method is about 20%. For example, for $\delta = 10^{-5}$, $M = 1$, 17 and 3 iteration steps are performed in stage 1 and 2, respectively. Moreover, comparing Table 5 with 6, the combined methods are more efficient if we have multiple zeros or a cluster of zeros.

7. Comments

For which M is method (M) the most efficient? This question is hard to answer because the arithmetical costs depend on $k_1, \ldots, k_m$; for the linear case $k_1 = \cdots = k_m = 1$, see [16]. We measured computer time in Section 6 and found that all methods are comparable. Which method is the most efficient in the example depends on the details in the computer realization.

The use of the Euclidean algorithm for polynomials of high degree causes numerical instability. Hence, if one observes stability problems (e.g., the convergence rate is not as expected, etc.) one should use exact arithmetic and keep the degrees of the polynomials involved as small as possible, i.e., use Grau's method, $M = 1$.

One advantage of simultaneous methods is that they avoid deflation, i.e., the division of polynomials by some well-approximating factor. Moreover, one might expect that the simultaneous methods are more robust with respect to poor initial data — for the linear case it is still conjectured that Durand–Kerner's method is almost globally convergent [11]. The combined methods, as introduced in Section 6, seem to be an appropriate tool for the practical computation of all polynomial roots. Nevertheless, the stopping criterions are expensive and the forming of clusters should be performed more flexibly and more adaptively, in particular using more information on multiplicities from the first stage (see, e.g. [1, 5, 8, 11, 12]).

We finally mention that a posteriori error estimates may be obtained, e.g., via an interpretation of Grau's method ($M = 1$) using companion matrices (cf. [4, Section 3]).

References

[1] O. Aberth, Iteration methods for finding all zeros of polynomial simultaneously, *Math. Comput.* **27** (1973) 339–344.
[2] G. Alefeld and J. Herzberger, On the convergence speed of some algorithms for the simultaneous approximation of the polynomial roots, *SIAM J. Numer. Anal.* **11** (1974) 237–243.
[3] D. Bini and L. Gemignani, On the complexity of polynomial zeros, *SIAM J. Comput.* **21** (1992) 781–799.
[4] C. Carstensen, On Grau's method for simultaneous factorization of polynomials, *SIAM J. Numer. Anal.* **29** (1992) 601–613.
[5] C. Carstensen, On quadratic-like convergence of the means for two methods for simultaneous root finding of polynomials, *BIT* **33** (1992) 64–73.
[6] C. Carstensen, On simultaneous factoring of a polynomial, *Internat. J. Comput. Math.* **46** (1992) 51–61.
[7] A. Cuyt and L. Wuytack, *Nonlinear Methods in Numerical Analysis* (North-Holland, Amsterdam, 1987).
[8] P. Fraigniaud, The Durand–Kerner polynomial root-finding method in case of multiple roots, *BIT* **31** (1991) 112–123.
[9] A.A. Grau, The simultaneous Newton improvement of a complete set of approximate factors of a polynomial, *SIAM J. Numer. Anal.* **8** (1971) 425–438.
[10] A.W. Householder, Generalizations of an algorithm of Sebastiano e Silva, *Numer. Math.* **16** (1971) 375–382.
[11] G. Kjellberg, Two observations on Durand–Kerner's root-finding method, *BIT* **24** (1984) 556–559.
[12] T. Miyakoda, Iterative methods for multiple zeros of a polynomial by clustering, *J. Comput. Appl. Math.* **28** (1989) 315–326.
[13] A.W. Nourein, Root determination by use of Padé approximants, *BIT* **16** (1976) 291–297.
[14] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* (Academic Press, New York, 1970).

[15] T. Pomentale, A class of iterative methods for holomorphic functions, *Numer. Math.* **18** (1971) 193–203.
[16] M.S. Petković, *Iterative Methods for Simultaneous Inclusion of Polynomial Zeros*, Springer Lecture Notes, Vol. 1387 (Springer, Berlin, 1989).
[17] T. Sakurai, T. Torii and H. Sugiura, A high-order iterative formula for simultaneous determination of zeros of a polynomial, *J. Comput. Appl. Math.* **38** (1991) 387–397.
[18] T. Sakurai, H. Sugiura and T. Torii, Numerical factorization of a polynomial by rational Hermite interpolation, *Numer. Algorithms* **3** (1992) 411–418.